

Desarrollo FrontEnd con JavaScript

s05 – Backbone.js : Vistas y routers

Model

Collection

Backbone.js

View

Router

Backbone.View

Backbone.View

```
var LibraryView = Backbone.View.extend({  
  el: '#library',  
  initialize: function() {  
    this.listenTo(this.collection, 'add', this.renderNewBook);  
  },  
  renderNewBook: function(newBook, collection, options) {  
    this.$el.find('ul.list').append('<li>' +  
newBook.get('title') + '</li>');  
  }  
});
```


Backbone.View

- Maneja la interacción de la aplicación con el usuario.
- Escucha los cambios realizados en el modelo o colección.

```
var LibraryView = Backbone.View.extend({  
  el: '#library',  
  events: {  
    'click #add-book': 'addBook'  
  },  
  addBook: function() {  
    this.collection.add({  
      title: this.$el.find('#book-title').val();  
    });  
  }  
});
```

Backbone.View

- El manejo de renderizado depende del usuario.
- Permite usar templates:
 - Underscore.js
 - Haml-js
 - Mustache.js
 - Handlebars.js

```
var LibraryView = Backbone.View.extend({
  el: '#library',
  template: _.template('<li><%= title %></li>'),
  render: function() {
    var self = this, content = '';

    this.collection.each(function(book) {
      content += self.template(book.toJSON());
    });

    this.$el.html(content);
  }
});
```

Backbone.Router

Backbone.Router

```
var AppRouter = Backbone.Router.extend({
  routes: {
    '': 'index'
  },
  index: function() {
    var myLibrary = new LibraryView({
      collection: new BookCollection()
    });

    myLibrary.render();
  }
});
```


Backbone.Router

- Conecta rutas con funciones que manejan el flujo de la aplicación.

```
var BookRouter = Backbone.Router.extend({
  routes: {
    'book/:id': 'bookDetail'
  },
  bookDetail: function(id) {
    var book = this.collection.get(id);
    alert(book.get('title'));
  },
  initialize: function(options) {
    this.collection = options.collection;
  }
});
```

Backbone.history

Backbone.history

- Inicializa la aplicación, empezando a capturar los eventos **hashchange**.
- **start()** debe ser llamado después de haber inicializado todos los routers de la aplicación.

```
var myLibrary = new BookCollection();

var appRouter = new AppRouter({
  collection: myLibrary
});

var libraryRouter = new BookRouter({
  collection: myLibrary
});

Backbone.history.start();
```